Reactive Self Collision Avoidance with Dynamic Task Prioritization for Humanoid Robots

Hisashi Sugiura, Michael Gienger, Herbert Janßen, Christian Goerick

2010

Preprint:

This is an accepted article published in International Journal of Humanoid Robotics (IJHR). The final authenticated version is available online at: https://doi.org/[DOI not available]

International Journal of Humanoid Robotics
Vol. 7, No. 1 (2010) 1–24

DOI: 10.1142/S0219843610001976

REACTIVE SELF COLLISION AVOIDANCE WITH DYNAMIC TASK PRIORITIZATION FOR HUMANOID ROBOTS

HISASHI SUGIURA*, MICHAEL GIENGER, HERBERT JANßEN
and CHRISTIAN GOERICK

Honda Research Institute Europe GmbH,
D-63073 Offenbach/Main, Germany
**hisashi.sugiura@honda-ri.de*

Received 25 January 2008
Revised 24 April 2009
Accepted 23 June 2009

We propose a self collision avoidance system for humanoid robots designed for interacting with the real world. It protects not only the humanoid robots' hardware but also expands its working range while keeping smooth motions. It runs in real-time in order to handle unpredictable reactive tasks such as reaching to moving targets tracked by vision during dynamic motions like e.g. biped walking.

The collision avoidance is composed of two important elements. The first element is reactive self collision avoidance which controls critical segments in only one direction — as opposed to other methods which use 3D position control. The virtual force for the collision avoidance is applied to this direction and therefore the system has more redundant degrees of freedom which can be used for other criteria. The other second element is a dynamic task prioritization scheme which blends the priority between target reaching and collision avoidance motions in a simple way. The priority between the two controllers is changed depending on current risk.

We test the algorithm on our humanoid robot ASIMO and works while the robot is standing and walking. Reaching motions from the front to the side of the body without the arm colliding with the body are possible. Even if the target is inside the body, the arm stops at the closest point to the target outside the body. The collision avoidance is working as one module of a hierarchical reactive system and realizes reactive motions. The proposed scheme can be used for other applications: We also apply it to realizing a body schema and occlusion avoidance.

Keywords: Collision avoidance; task priority; occlusion; redundant control.

1. Introduction

One of the most desirable properties of motion generation is to reach the target while dealing with many constraints which interfere with the target reaching motions. For example, the range of joints restricts the overall robot's working range and the limitations of actuators restrict the velocity and acceleration of joints. Collisions between robot segments also restrict robot motions. In particular, the motion generation should solve these problems in real-time for reactive motions in an unpredictable environment.

Research of collision avoidance has been carried out for years as a part of trajectory planning. Lozano-Pérez *et al.* have proposed a configuration space approach.[17] In this space, the robots' position and orientation are characterized as a reference point which makes it easier to generate collision free trajectories compared to using actual 3D space.

Kuffner *et al.* have proposed a fast collision detection[12] and trajectory planning algorithm which allows a simultaneously collision free and dynamic balancing motion on the humanoid robot H7.[11] The trajectory planning uses an algorithm called RRT-Connect in configuration space.

Although these methods generate optimal trajectories which satisfy criteria in static environments, they are difficult to be applied in non-predetermined environments which may change momentarily because trajectories may need to be regenerated every time slice. In particular, on robots which have many degrees of freedom (DOFs) such as humanoid robots this method causes increasing computational costs.

Some methods for reactive collision avoidance - called "real-time" collision avoidance by some authors - have been proposed. Our method presented in this paper belongs to this category. There are mainly two important elements to consider for reactive collision avoidance. One is how to avoid collisions and the other one is how to prioritize between target reaching motions and collision avoidance motions.

Bicho *et al.* have proposed a method using attractor dynamics to determine the direction of avoidance for a mobile robot.[1] Regarding articulated mobile robots, Khatib has proposed a method based on potential fields: target and obstacles are represented by attractive and repulsive potentials respectively.[10] The robot follows this field and reaches to the target in real time. Brock *et al.* have proposed the "elastic strip" framework in order to modify planned trajectories. They have applied it to a wheeled robot with one arm.[2] Seto *et al.* have proposed the concept of a "Representation of Body by Elastic Elements" which generates virtual forces. They have applied it to a wheeled robot which has two arms with seven joints respectively.[27]

These methods have been applied to redundant arms and they are efficient. However, they use three DOFs to move a critical segment. We have already proposed a collision avoidance method using nullspace optimization criteria[6] and task intervals[29] which uses closest points defined by shortest distance between segments instead of control points. However our previously proposed method moves a 3D closest point which also uses three DOFs.

Humanoid robots need higher redundancy in order to solve many criteria. Target reaching motions usually need a position and sometimes an orientation which uses up to six DOFs in total. Additionally some joints cannot always be freely used due to limitations such as joint ranges, joint velocity, joint acceleration and singularities. Therefore collision avoidance should use a minimum of DOFs in order not to violate limitations or criteria of the motion control even if robots have redundant DOFs.

Research on prioritizing different tasks, has mainly been developed as part of redundant control.[3,16,18,21] Hanafusa *et al.* have introduced a concept "tasks with priority" which can prioritize many tasks using nullspace optimization criteria.[8] They have applied the method to target reaching and collision avoidance motions on a redundant articulated robot. This method allows to reach for targets while avoiding collisions. Siciliano proposed a recursive extension[28] and Mansard realized smooth transitions when switching tasks[20] based on the method.

Another class of methods is using a weighting matrix — sometimes called regularization or damping matrix — which determines the sensitivity of each task. The first publications were Nakamura *et al.*[22] who proposed a method called Singularity Robustness (SR-Inverse) and Wampler[34] who proposed a method called Damped Least Square (DLS) method. Tsuji *et al.*[32] also proposed a similar method but with a different scheme to derive task prioritization.[26,27] Note that if the weighting matrix method is used for collision avoidance, a different way is needed to handle singularities.

In this paper, we first propose a self collision avoidance scheme which minimizes the influence to regular motions and other criteria. The method does not use three DOFs but uses only one DOF by generating a collision avoidance motion only in the critical direction between closest segments. Second, we propose a conflict resolution which dynamically changes the priority between collision avoidance and target reaching motions by means of blending both controllers in a simple way. Our method changes the priority between target reaching and collision avoidance motion smoothly and automatically depending on a level of risk. It does not completely switch off either motions unless in the case of extreme situations by employing both task space and nullspace.

We have already presented a framework for a behavior control system, so-called *ALIS* (Autonomous Learning and Interaction System)[7] which enables humanoid robots to interact with the real world. It comprises visual saliency, sound localization, online learning of visual proto-objects and body control. The collision avoidance is integrated in this system and works cooperatively with the whole body motion control scheme described by Gienger *et al.*[6] For ALIS, self collision avoidance is crucially necessary because arbitrary target commands may be guaranteed during interaction.

A preliminary version of our self collision avoidance and first results have been presented before.[30] Here, we present the full collision avoidance which avoids multiple objects with smooth motions when target reaching motions and collision avoidance motions are blended. Additionally, we present concepts of a body schema and an occlusion avoidance.

This paper is organized as follows. In the next section we describe the overall concept of our system. The system comprises the whole body motion controller and the collision avoidance controller. The whole body motion controller is described in Sec. 3. Section 4 describes the distance computation which is necessary for the collision avoidance. The novelties of this paper, the collision avoidance and the dynamic task prioritization are described in Secs. 5 and 6, respectively.

Section 7 presents how the system is implemented and Sec. 8 shows some experimental results on the simulator and the robot. Finally we discuss and conclude in Secs. 9 and 10, respectively.

2. Overall Concept

The fundamental concept of dynamic task prioritization is not to switch but to blend between the collision avoidance controller and the whole body motion controller continuously. The prioritization is simple and does not switch controllers unless in the case of extreme situations.

For collision avoidance, we define two controllers, the whole body motion controller and the collision avoidance controller.

We define

$$\dot{\boldsymbol{q}} = \{1 - f(d)\}\dot{\boldsymbol{q}}_{wbm} + f(d)\dot{\boldsymbol{q}}_{ca}, \tag{1}$$

where $\dot{\boldsymbol{q}}$ is the resulting joint velocity vector of the robot, $f(d)$ is the blending coefficient and d is the distance between closest segments. $\dot{\boldsymbol{q}}_{wbm}$ and $\dot{\boldsymbol{q}}_{ca}$ are resulting joint velocity vectors for the whole body motion controller and the collision avoidance controller, respectively. The details of the whole body motion controller, the collision avoidance controller and the prioritization are described in following sections.

3. Whole Body Motion Control

We use a redundant control scheme for whole body motion control[6] so that the robot which has redundant DOFs reaches its targets. The redundant control is described by

$$\dot{\boldsymbol{q}}_{wbm} = \boldsymbol{J}^{\#}_{wbm}(\boldsymbol{q})\dot{\boldsymbol{r}}_{task} + \boldsymbol{N}_{wbm}\boldsymbol{\xi}_{wbm}, \tag{2}$$

where $\dot{\boldsymbol{q}}_{wbm}$ is the joint velocity vector, $\boldsymbol{J}^{\#}{}_{wbm}(\boldsymbol{q})$ is the pseudo inverse of task Jacobian matrix $\boldsymbol{J}_{wbm}(\boldsymbol{q})$ and $\dot{\boldsymbol{r}}_{task}$ is the task velocity vector. The matrix $\boldsymbol{N}_{wbm}$ which maps an arbitrary vector $\boldsymbol{\xi}_{wbm}$ into the nullspace is written

$$\boldsymbol{N}_{wbm} = \boldsymbol{I} - \boldsymbol{J}^{\#}_{wbm}(\boldsymbol{q})\boldsymbol{J}_{wbm}(\boldsymbol{q}). \tag{3}$$

We project gradients of a potential function into the nullspace as

$$\boldsymbol{\xi}_{wbm} = \boldsymbol{\xi}_{wbm,ca} + \boldsymbol{\xi}_{wbm,jla}, \tag{4}$$

$$\boldsymbol{\xi}_{wbm,ca} = -\alpha_{ca}[\boldsymbol{J}_{ca}(\boldsymbol{q})|_{row,y}]^T(d_{safe} - d), \tag{5}$$

where $\boldsymbol{\xi}_{wbm,ca}$ is a gradient of a collision avoidance function, $\boldsymbol{\xi}_{wbm,jla}$ is a gradient of a joint limit avoidance function.[6] $\boldsymbol{J}_{ca}(\boldsymbol{q})$ is the Jacobian matrix for collision avoidance between closest points $\boldsymbol{P}_{cp1}$ and $\boldsymbol{P}_{cp2}$. $\boldsymbol{J}_{ca}(\boldsymbol{q})|_{row,y}$ is the row vector which is y element of $\boldsymbol{J}_{ca}(\boldsymbol{q})$. d_{safe} is a safety distance which is arbitrary but must be sufficiently large so that $d_{safe} - d$ is always positive and α_{ca} is the step width.

The whole body motion control projects joints for the target reaching in task space and the collision avoidance works in nullspace.

4. Distance Computation

For our collision avoidance scheme, it is necessary to compute distances and closest points between segments, i.e. the physical links separated by joints. Many methods for this have been proposed not only in robotics but also in computer games based on accurate models using, for instance, convex polyhedra.[4,5,12,15,33] These models with high level of detail contribute to larger robot working ranges.

However, it is computationally expensive to compute distances and closest points for all possible segment pairs of humanoid robots with detailed models especially with real-time constraints.

We therefore define the collision model with shape primitives that can be computed faster (see in Fig. 1). Each segment (*Head*, *Right Shoulder* and so on) is composed of spheres or sphere swept lines (capped cylinders) in order to cover the shape of the robot.[14] Most of the segments are composed of one primitive object, but the body and the chest use multiple primitive objects. Although the real shape of some segments — in particular the body segment — is only coarsely approximated by the primitives, the primitives are in all cases the minimal enclosing ones. We compute distances and closest points between the segments which are potentially colliding based on this model.

With these shape primitives the working range of the robot is smaller than for finer models, however, the working range in front of the body — which is most important for manipulation tasks — is *not* limited by this model.

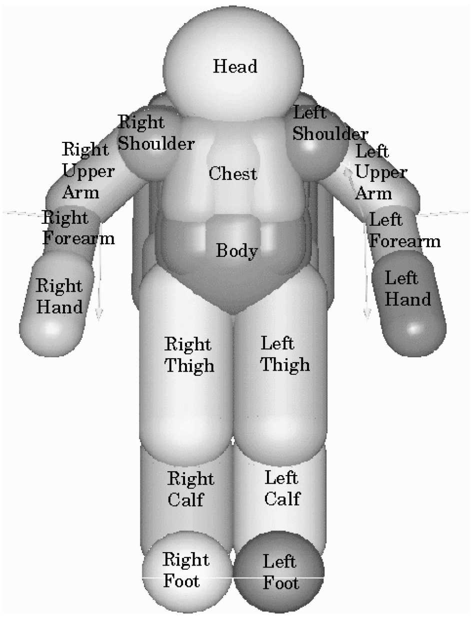


Fig. 1. The employed collision model is composed of 17 segments. Each segment is composed of one or several sphere swept lines or spheres.

The most critical situations are when a direct trajectory passes through a segment or when a given target is inside a body. Both cases are shown in Sec. 8.

5. Reactive Self Collision Avoidance

We use separate controller for collision avoidance with virtual forces. The major function of collision avoidance is to keep a distance between the two closest segments of the robot. As we mentioned in Sec. 1 the collision avoidance methods which have been proposed so far move 3D segment positions to other safer 3D positions. For example, Seto *et al.* define a 3D control point and move it to safer position.[27] Sentis *et al.* use the constraint force vector to move the closest segment.[25]

However, it is sufficient to separate two close segments in the one critical direction which can be managed with only one DOF. For this purpose, we define a collision avoidance coordinate system so that one of its axes aligns to the critical avoidance direction. The closest segments are separated along this axis, that is, the collision avoidance controller moves the segments in only one direction. Therefore the other degrees of freedom can be used more for other criteria than moving 3D segment positions.

5.1. *Collision avoidance coordinate system*

We define a collision avoidance coordinate system so that one of the closest points $\boldsymbol{P}_{cp1}$ is the origin of the coordinate system and the y axis passes through the closest point $\boldsymbol{P}_{cp2}$. The direction of y axis is the critical direction of the collision avoidance. The other axes are arbitrary. Figure 2 shows an example of the collision avoidance coordinate system. In this case, $\boldsymbol{P}_{cp1}$ and $\boldsymbol{P}_{cp2}$ are located on the *Right Forearm* and the *Body* respectively.

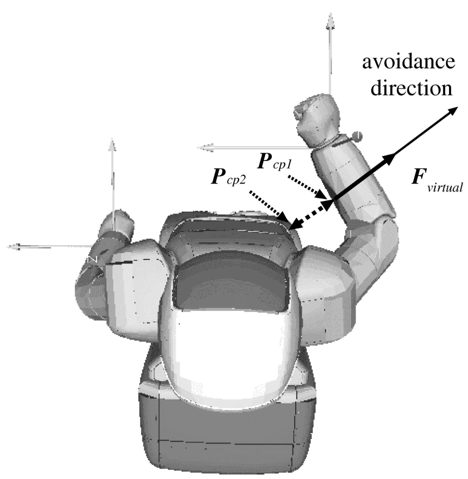


Fig. 2. The closest points, the virtual force and the collision avoidance coordinate system (gray arrows).

5.2. *Virtual force*

We use a virtual repulsive force in order to push a pair of potentially colliding segments away from each other in the collision avoidance coordinate system. The virtual force $\boldsymbol{F}_{virtual}$ is applied to $\boldsymbol{P}_{cp1}$ in the avoidance direction which is aligned to the y axis as discussed before. $\boldsymbol{F}_{virtual}$ is written

$$\boldsymbol{F}_{virtual} = \begin{cases} (0 \;\; k(d_a - d) \;\; 0)^T & \text{if} \quad d < d_a, \\ \mathbf{0} & \text{otherwise}, \end{cases} \tag{6}$$

$$d = |\boldsymbol{P}_{cp1} - \boldsymbol{P}_{cp2}|, \tag{7}$$

where k is a positive constant and d_a defines the boundary of the volume (so-called *yellow zone*) in which the joint velocity vector is blended according to Eq. (1).

5.3. *Collision avoidance controller*

The collision avoidance control also uses a redundant control scheme for two motions: collision avoidance motion in task space and target reaching motion in nullspace.[6] Employing a potential function, the joint velocity vector $\dot{\boldsymbol{q}}_{ca}$ is computed as

$$\dot{\boldsymbol{q}}_{ca} = [\boldsymbol{J}_{ca}(\boldsymbol{q})|_{row,y}]^{\#} F_{virtual} + \boldsymbol{N}_{ca}\boldsymbol{\xi}_{ca}, \tag{8}$$

$$\boldsymbol{N}_{ca} = \boldsymbol{I} - \boldsymbol{J}_{ca}(\boldsymbol{q})^{\#}\boldsymbol{J}_{ca}(\boldsymbol{q}). \tag{9}$$

Matrix $\boldsymbol{J}_{ca}(\boldsymbol{q})$ is the collision avoidance Jacobian between points $\boldsymbol{P}_{cp1}$ and $\boldsymbol{P}_{cp2}$ on the collision avoidance coordinate system. There must be at least one joint between the segments. Column vector $[\boldsymbol{J}_{ca}(\boldsymbol{q})|_{row,y}]^{\#}$ is a pseudo inverse vector of the row vector $\boldsymbol{J}_{ca}(\boldsymbol{q})|_{row,y}$. Since the collision avoidance system needs to separate closest points only in one direction, the Jacobian column vector $[\boldsymbol{J}_{ca}(\boldsymbol{q})|_{row,y}]^{\#}$ is used instead of the Jacobian matrix $\boldsymbol{J}^{\#}_{ca}(\boldsymbol{q})$. Matrix $\boldsymbol{N}_{ca}$ maps a vector of a gradient vector $\boldsymbol{\xi}_{ca}$ in the nullspace of the motion and scalar $F_{virtual}$ is expressed as

$$F_{virtual} = |\boldsymbol{F}_{virtual}| = \boldsymbol{F}_{virtual}|_y, \tag{10}$$

where $\boldsymbol{F}_{virtual}|_y$ is the y element of $\boldsymbol{F}_{virtual}$.

Thus the collision avoidance effectively affects only one DOF. The gradient of the potential function $\boldsymbol{\xi}_{ca}$ is explained in the following.

5.4. *Potential function*

To reach the target, we project the gradient of a target distance function $H_t(r)$ to the nullspace of the movement with a weighting matrix $\boldsymbol{W}_t$:

$$H_t(\boldsymbol{r}) = \frac{1}{2}(\boldsymbol{r} - \boldsymbol{r}_t)^T \boldsymbol{W}_t(\boldsymbol{r} - \boldsymbol{r}_t). \tag{11}$$

where $\boldsymbol{r}$ is the current task vector and $\boldsymbol{r}_t$ is the target task vector. Note that $\boldsymbol{r}$ and $\boldsymbol{r_t}$ can theoretically include both hands, the head and the orientation of the robot, or any combination of those. Therefore even if the whole body motion controller does not effect, that is $f(d) = 1$ in Eq. (1), target reaching is realized in the nullspace. Let α_t be a step width, then gradient vector $\boldsymbol{\xi}_{ca}$ is expressed by

$$\boldsymbol{\xi}_{ca} = -\alpha_t \left(\frac{\partial H_t(\boldsymbol{r})}{\partial \boldsymbol{q}} \right)^T = -\alpha_t \boldsymbol{J}_{wbm}(\boldsymbol{q})^T \nabla \boldsymbol{H}(\boldsymbol{r}). \tag{12}$$

6. Dynamic Task Prioritization

We have defined two controllers, the whole body motion controller and the collision avoidance controller. The collision avoidance controller maps the collision avoidance motion into task space and the target reaching motion into nullspace. Vice versa the whole body motion controller maps the target reaching motion into the task space and the collision avoidance motion into the nullspace. The priority of the two motions in both controllers is always fixed.

However, for the reactive collision avoidance, it is necessary to change the priority between target reaching and collision avoidance motions dynamically with low computational burden. If the trajectory of a target reaching motion is far from collisions, the collision avoidance motions should not disturb it. But if a segment comes closer to another segment, the collision avoidance motion should have higher priority. We proposed our concept in Eq. (1) as

$$\dot{\boldsymbol{q}} = \{1 - f(d)\}\dot{\boldsymbol{q}}_{wbm} + f(d)\dot{\boldsymbol{q}}_{ca}.$$

Function $f(d)$ plays the role of changing the priority between two joint velocity vectors, $\dot{\boldsymbol{q}}_{wbm}$ and $\dot{\boldsymbol{q}}_{ca}$ which are outputs of symmetrical equations Eq. (2) and Eq. (8) integrated with the function $f(d)$. It is the level of risk of collisions as,

$$f(d) = \begin{cases} 1 & \text{if } d \leq d_b, \\ \dfrac{d - d_a}{d_b - da} & \text{else if } d_b < d \leq d_a, \\ 0 & \text{else,} \end{cases} \tag{13}$$

where a distance d_b determines a so-called *orange zone* which is always smaller than d_a. If the closest segment is further apart, the whole body motion control $\dot{\boldsymbol{q}}_{wbm}$ as in Eq. (2) has full control of the motion. On the other hand, if the closest segment moves into this zone, the collision avoidance control dominates the motion. If d is between d_b and d_a, both the collision avoidance control and the whole body motion control affect the robot motion weighted with the function $f(d)$.

Note that the collision avoidance motion always affects the robot motion even if $d > d_a$ by means of the nullspace of the whole body motion control. The target reaching motion is also affected by the nullspace of the collision avoidance control even if $d < d_b$.

7. Implementation

Based on the outlined control mechanism, we have implemented the collision avoidance on our humanoid robot Asimo.

7.1. *Emergency stop*

Our system uses distance computations not only for collision avoidance but also for collision detection which has been commonly used for robots in order to activate an emergency stop mechanism. If the closest distance between segments becomes less than d_r (so-called *red zone*), then the emergency stop will freeze the robot. All distance computations are done on the embedded computers, so the robot's does not depend on the network connection to external computers.

7.2. *Motion priority between collision avoidance*

Each virtual force vector $\boldsymbol{F}_{virtual}$ in Eq. (6) for the right and left arm is computed depending on each distance d. When the right and the left arm are very close, the collision avoidance handles the motion according to a task-depended "motion priority."

If the motion priority of the right arm is higher than the left one, then only the left arm avoids the right one and the right one moves to the target without being influenced by the collision avoidance.

7.3. *Closest points and virtual forces*

According to Fig. 2, the closest points are computed for one segment pair, but actually our algorithm takes all potentially colliding segment pairs into account, so that $\boldsymbol{F}_{virtual}$ and $\boldsymbol{P}_{cp1}$ or $\boldsymbol{P}_{cp2}$ do not produce discontinuities. Although this does not eliminate discontinuities completely it proved to be a very big practical improvement in terms of motion smoothness.

If a closest distance d_{ij} between a segment S_i and the other segment S_j is smaller than d_a, a virtual force vector $\boldsymbol{F}_{ij}$ is applied to the arm segment:

$$\boldsymbol{F}_{ij} = k(d_a - d_{ij})\boldsymbol{e}_{f,ij}, \tag{14}$$

$$d_{ij} = |\boldsymbol{P}_{cp1,ij} - \boldsymbol{P}_{cp2,ji}| \tag{15}$$

$$\boldsymbol{e}_{f,ij} = \frac{1}{d_{ij}}(\boldsymbol{P}_{cp1,ij} - \boldsymbol{P}_{cp2,ji}), \tag{16}$$

where $\boldsymbol{P}_{cp1,ij}$ and $\boldsymbol{P}_{cp2,ji}$ are the closest points on S_i and S_j, respectively. The overall virtual force vector $\boldsymbol{F}^{all}_{virtual,i}$ which is applied from $S_0, S_1, \ldots, S_{n-1}$ to S_i is

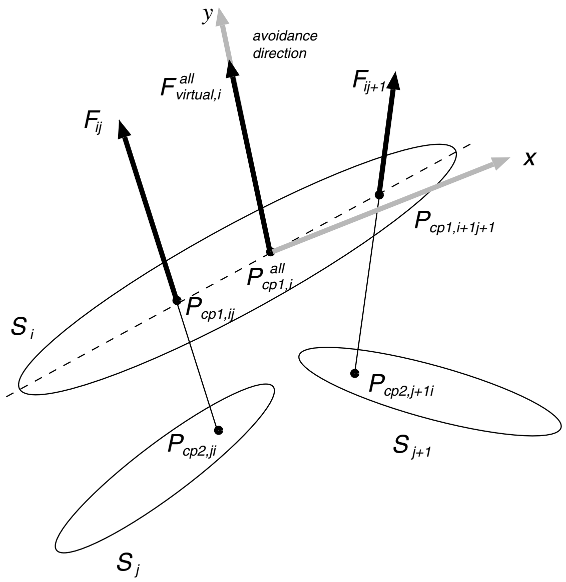


Fig. 3. The overall closest point, the virtual force and the collision avoidance coordinate system (gray arrows).

computed using the internal division with $\boldsymbol{F}_{ij}$:

$$\boldsymbol{F}^{all}_{virtual,i} = \frac{\displaystyle\sum_{j=0}^{n-1} |\boldsymbol{F}_{ij}|\boldsymbol{F}_{ij}}{\displaystyle\sum_{j=0}^{n-1} |\boldsymbol{F}_{ij}|}. \tag{17}$$

The overall closest point $\boldsymbol{P}^{all}_{cp1,i}$ on S_i where $\boldsymbol{F}^{all}_{virtual,i}$ force is applied is expressed with the closest points of segment pairs $\boldsymbol{P}_{cp1,ij}$ as,

$$\boldsymbol{P}^{all}_{cp1,i} = \frac{\displaystyle\sum_{j=0}^{n-1} |\boldsymbol{F}_{ij}|\boldsymbol{P}_{cp1,ij}}{\displaystyle\sum_{j=0}^{n-1} |\boldsymbol{F}_{ij}|}. \tag{18}$$

The origin of the collision avoidance coordinate system is $\boldsymbol{P}_{cp1}$ and the y axis is aligned to $\boldsymbol{F}_{virtual}$. $\boldsymbol{P}^{all}_{cp1,i}$ can be used instead of $\boldsymbol{P}_{cp1}$ and $\boldsymbol{F}^{all}_{virtual,i}$ can be used instead of $\boldsymbol{F}_{virtual}$ after being transformed to the collision avoidance coordinate system. An example is shown in Fig. 3.

8. Experiments

In this section, we describe the experiments and the results which have been obtained on our humanoid robot. The method is applied to all potentially colliding segments. The leg segments are not controlled by the system in order not to disturb the walking and balancing system which controls the legs exclusively, however, the arm segments avoid leg segments according to applied virtual forces from relevant leg segments. In other words, the motion priority of the leg segments is higher than the arm segments.

In the experiments, we first use some basic examples in order to show that the proposed scheme works correctly. Second we test with more complex and realistic examples. Further we have tested the presented scheme running within a behavior control system (ALIS[7]). Finally, we describe some applications.

8.1. *Humanoid robot ASIMO*

Experiments have been carried out on our humanoid robot ASIMO[9] which we describe with 21 DOFs in total comprising five DOFs for each arm (three joints on the shoulder, one on the elbow and one on the wrist) and six DOFs to describe the virtual link between heel and upper body, three DOFs for the heel coordinate and two DOFs for head motions.

The overall system is composed of the collision avoidance control, the distance computation and the whole body motion. It is depicted in Fig. 4.

The distance computations for the collision avoidance are done with all segment pairs which can collide as shown in Fig. 1. For instance, the distances from *Right Forearm* and *Right Hand* against *Head*, *Left Shoulder*, *Left Upper Arm*, *Left Forearm*, *Left Hand*, *Chest*, *Body* and *Right Thigh* are computed respectively. The *Left Forearm* and *Left Hand* are computed in the same manner. The distance thresholds have been set to $d_a = 40\,\text{mm}$, $d_b = 10\,\text{mm}$ and $d_r = 5\,\text{mm}$. The controller for the collision models in Fig. 1 is operating on the five joints (three shoulder joints, one elbow joint and one wrist joint) for each arm.

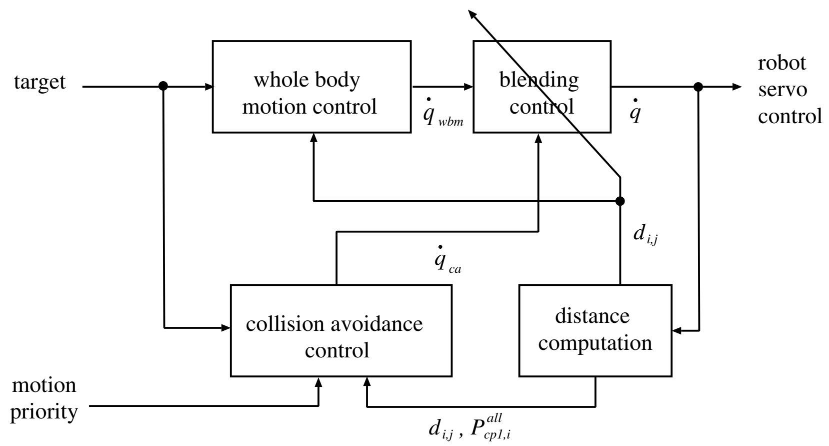


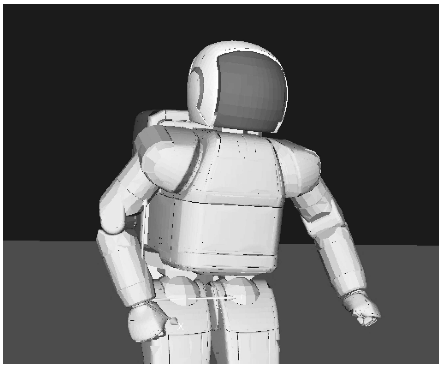
Fig. 4. Our system for collision avoidance, the distance computation and the whole body motion control. Both *target* and *motion priority* are given by higher level modules.

If a new target is commanded, a linear trajectory between the current and the new target is generated. All computations except vision processing and high level behavior control are performed on the robot's embedded computer. The sampling time for computations of the total control system including the distance computations and the collision avoidance is 5 m sec.

8.2. *Basic examples*

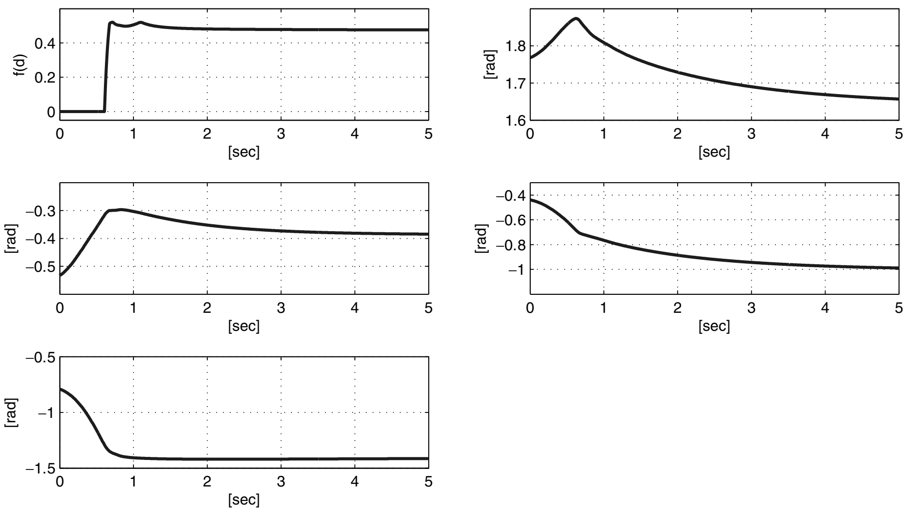
8.2.1. *The reaching target is inside the body*

The target of the right arm is inside the body. When the collision avoidance is deactivated, the lower arm collides with the body, as shown in Fig. 5(a). Figure 5(b)



(a) Simulation without Avoidance

(b) Robot with Avoidance



(c) The blending coefficient $f(d)$ and joint angles

Fig. 5. Example of a target that is inside the body. In simulation without collision avoidance, the right hand collides with the right leg in (a). This doesn't happen in (b). (c) shows the blending coefficient $f(d)$ (left top) and three shoulder joints (from the right top to the right of the second row) and the elbow joint (bottom left) which are used for the collision avoidance in this example.

shows that the arm motion stops at the side of the body with collision avoidance activated on the real robot. Figure 5(c) shows the blending coefficient $f(d)$ and the joint angles which are used for the collision avoidance of this example. The joints change continuously and do not oscillate while the whole body motion controller and the collision avoidance controller are being blended.

8.2.2. *The trajectory between the current and the target position is passing through the robot's body*

In Fig. 6, segments of the robot lie on the trajectory between the current and the target position, but the target is outside the body. This is a typical case in which the robot cannot reach the target and restricts the working range of the robot without the collision avoidance. The collision avoidance pushes the arm outward by means of the virtual force while the arm limbs would collide with the body or the leg.

Figure 7(a) illustrates the trajectory of the right wrist with and without the collision avoidance of the example shown in Fig. 6. The trajectory without the collision avoidance violates ASIMO's body segment but the trajectory with the collision avoidance moves around the body to avoid collisions. While the trajectory traces the boundary of the yellow zone of the body segment, the virtual force is constant in Fig. 7(b).

Figure 8 compares the collision avoidance controller which uses one DOF and three DOFs. In this example, both controllers use four DOFs which joints are between the body segment and the right forearm segment. The former controller has three DOFs for reaching the given target which are mapped into the nullspace. On the other hand, the latter controller has only one DOF for reaching the given target in nullspace. Therefore, the controller which uses one DOF reaches the target

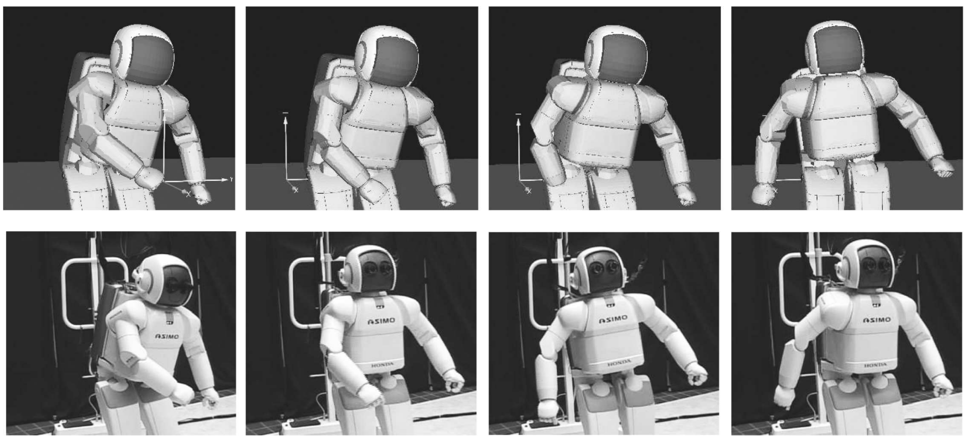
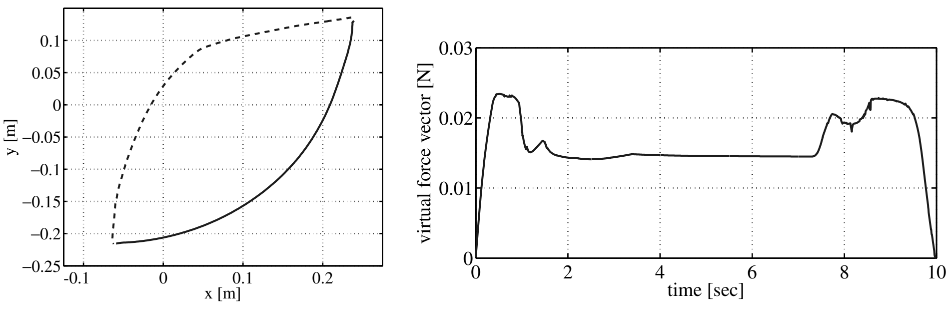


Fig. 6. Series of postures of the body avoidance. The same motion target is given both in the simulation without avoidance (top) and on the robot with avoidance (bottom) from the left to the right. Left to right sequential time frames from initial position to final position are shown.



(a) Trajectory on horizontal plane (b) Virtual force

Fig. 7. (a): The solid line shows the right wrist trajectory from the top right to the bottom left with collision avoidance (on ASIMO) while the dashed line shows the case without collision avoidance (on the simulator). (b): The corresponding virtual force between forearm and body. The discontinuities are caused by a conflict between the joint limit and the collision avoidance.

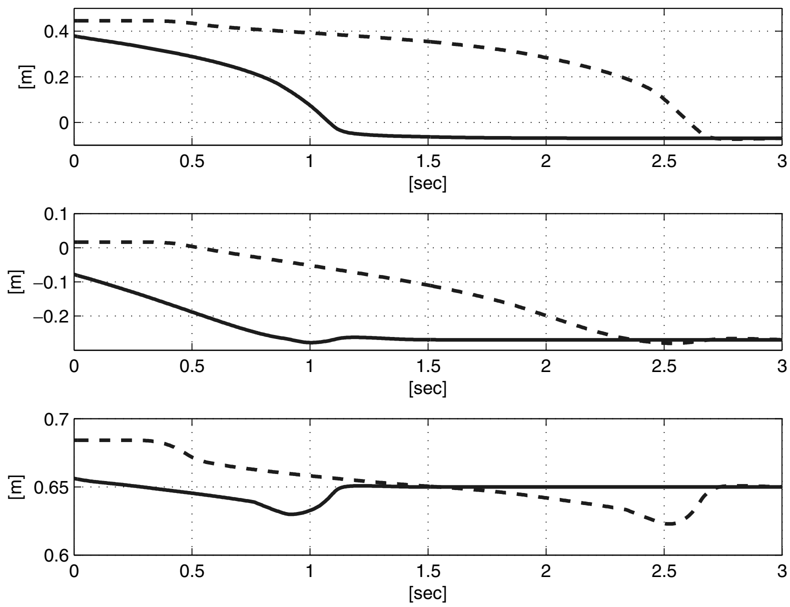


Fig. 8. The trajectory of the right hand x, y and z position with respect to the world coordinate system from the top to the bottom. The solid lines indicate the trajectories using our collision avoidance system which uses one DOF for avoidance and the dashed lines indicates using a collision avoidance which uses three DOFs for avoidance. Note that the start and the target positions are not identical to Fig. 7 in order to highlight the effectiveness of our scheme.

faster than the other controller. Figure 9 shows the blending coefficient $f(d)$ and the joints which are used for the collision avoidance of the former controller. The joints also change continuously and do not oscillate when both controllers are being blended.

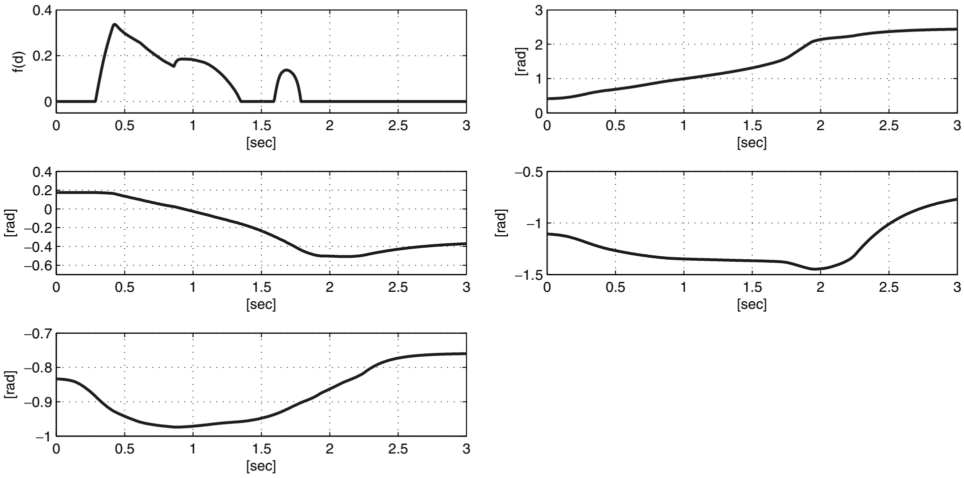


Fig. 9. The blending coefficient $f(d)$ (left top) and joint angles for the motion discussed in Fig. 8. The joints are three joints on the shoulder (from the right top to the right of the second row) and one joint on the elbow of the right arm (bottom left).

8.3. *Complex examples*

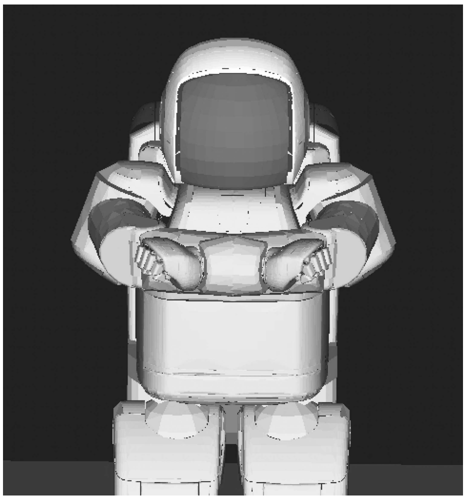
8.3.1. *Arms' targets lead to arm-arm-collision*

The targets for the arms are static 3D positions in front of the body. They are chosen in such a manner that the arms have to cross and that the arms would collide without collision avoidance. Figure 10 shows an example. The final posture of the robot is close to the target commands but with minimum distance d_b between the forearms.

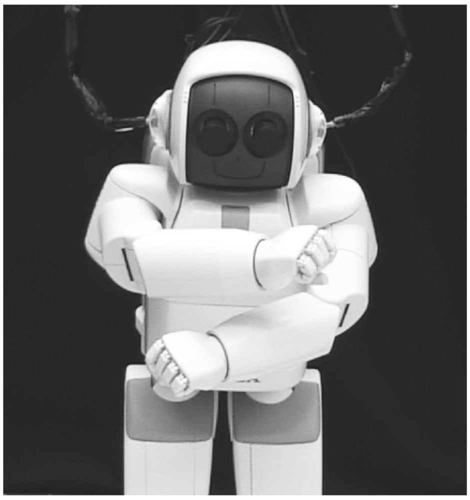
8.3.2. *The target is temporally inside the body while walking*

The collision avoidance also works while walking as illustrated in Fig. 11. The arm target in absolute coordinates is in front of the robot when it starts to walk in Fig. 11(a). The target for the walking is also in front of the robot but farther away. The robot's arm reaches its target as seen in Fig. 11(b). However, the target for the leg position is still farther away, so the robot continues walking. At some time the arm target is behind the robot (Fig. 11(c)) and the collision avoidance prevents the arm from penetrating the body. When the robot stops to walk, (Fig. 11(d)), the collision avoidance still affects the robot's motion seamlessly.

Note that as mentioned already the leg position is exclusively controlled by the walking and balancing controller and is not influenced by the collision avoidance, but vice versa the collision avoidance uses the leg position to calculate virtual forces on the arms.

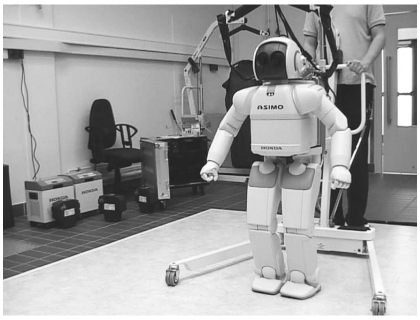


(a) Simulation without Avoidance

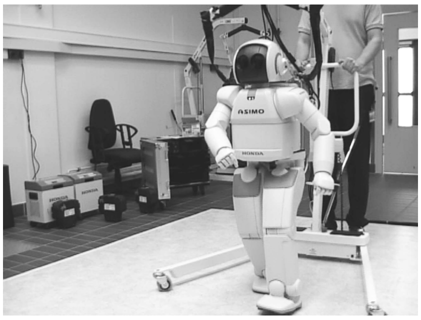


(b) Robot with Avoidance

Fig. 10. Example of arm collision avoidance. The motion without the collision avoidance on the simulator and with the collision avoidance on the robot.

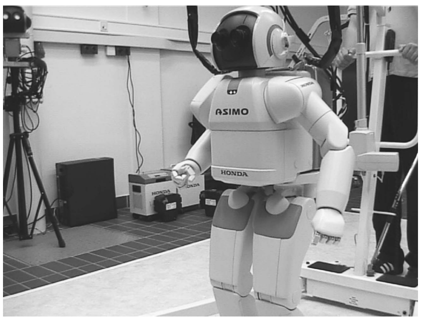


(a) Start walking



(b) Reach arm target

(c) Avoid while walking



(d) Stop walking

Fig. 11. Series of postures of the collision avoidance while walking. The target for the right arm is 80cm in front of the initial body position, and the robot is commanded to walk forward 1.5 m.

8.4. *Integrated to the interactive system*

We tested the collision avoidance in a reactive scenario that is a typical example for unpredictable motions in a behavior control system. One example of the scenario is shown in Fig. 12. A human holds two objects in front of the robot. The object positions are measured by a stereo camera vision system. The robot continuously points to the objects. The frame rate of the images is about 10 Hz. The robot has to continuously point and avoid collisions in real-time.

When two targets come into the range of the cameras, the robot points to them depicted in Fig. 12. In the scenario Asimo sometimes temporally "loses" an object, leading to only one hand pointing. Finally, the robot stops pointing because the human has crossed his arms leading to a target configuration that causes a closest hand distance of almost d_b and thus $f(d)$ is almost one in Eq. (1). Figure 13 shows the hand status and the closest distance between the arms and the hands for a longer interaction sequence. When the arm is not given a target (the hand status is 'no target'), the arm motions are determined by the criteria which are mapped

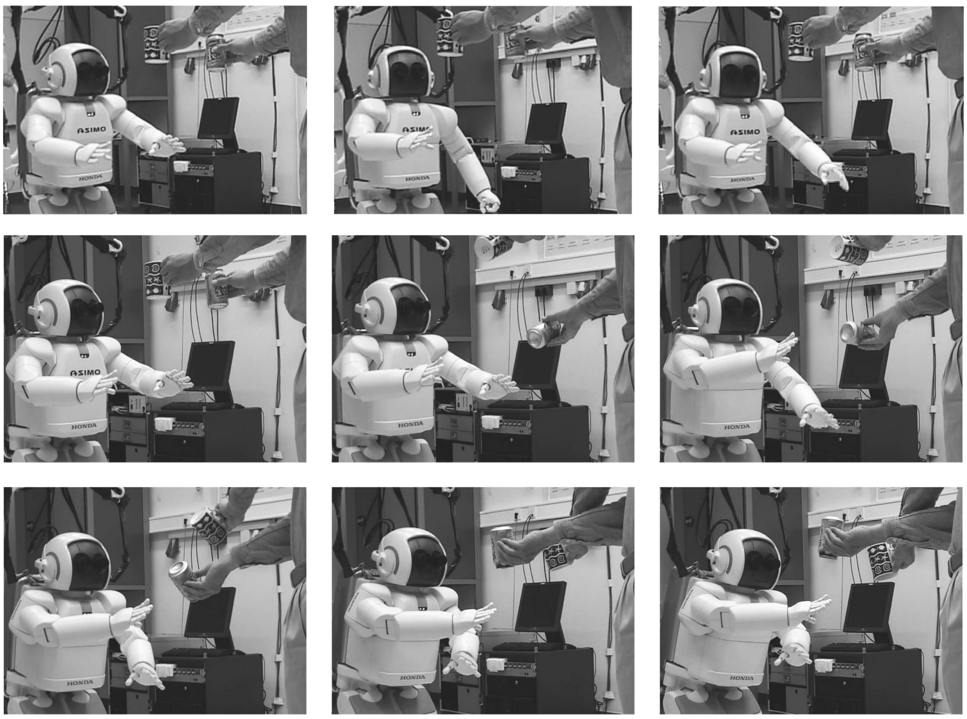


Fig. 12. This is an example of an reactive motion based on vision. The robot tracks two objects (a cup and a can) which are in the human's hands. When the robot loses one of the targets, the robot retracts its left hand (top-center) and the status of the left hand is "no target." Then both targets move counter clockwise in an arc from the robot's point of view and the robot tracks them. Finally, human's arms collide but the robot's arms don't because of the collision avoidance.

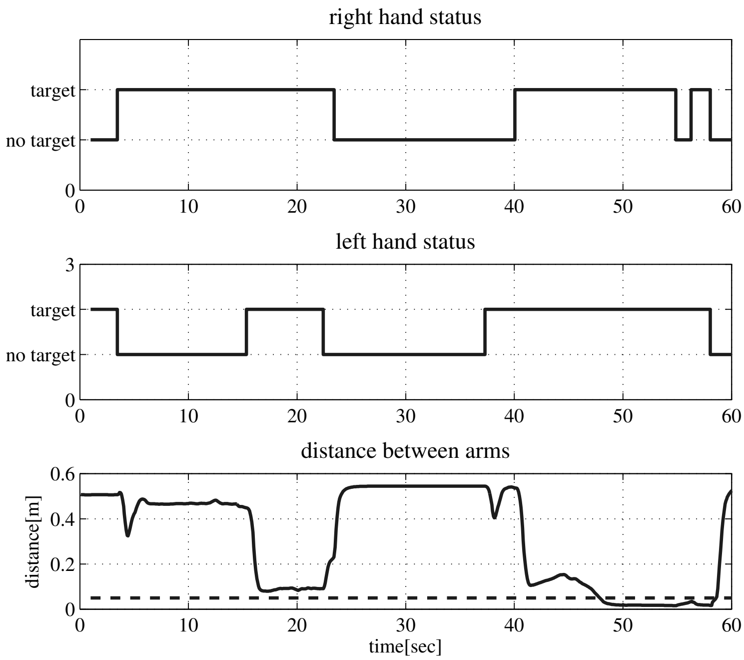


Fig. 13. The status of hands and the distance between arms. If the hand status is 'no target', no command is given to the hand. If it is 'given', a command is given to the respective hand. The dashed line represent the yellow zone boundary.

into nullspace in Eq. (2). ASIMO's arms move to the target which is shown to its cameras. When two targets are found, both arms move to their targets. From 40 sec to 55 sec, ASIMO continuously tracks two targets with two arms and finally the targets' 3D positions result in almost colliding arm postures. But the collision avoidance works from 48 sec to 58 sec (hand distance inside yellow zone) to prevent both arms from colliding.

The collision avoidance is running all the time, however, it does not disturb motions most of the time. We have already tested the behavior control system for hundreds of hours in different interaction scenarios during ongoing research work without any collisions.

8.5. *Application*

8.5.1. *External objects are considered as a part of the body (body schema)*

External objects can be considered as additional segments that the robot avoids. For instance, in Fig. 14, the robot grasps an object that can be considered as an additional segment. The robot avoids this segment as if it was one of the robot's segments. This corresponds to an extension of the robot's body schema.[24] In this example, the motion priority of the object is higher than the right hand segment.

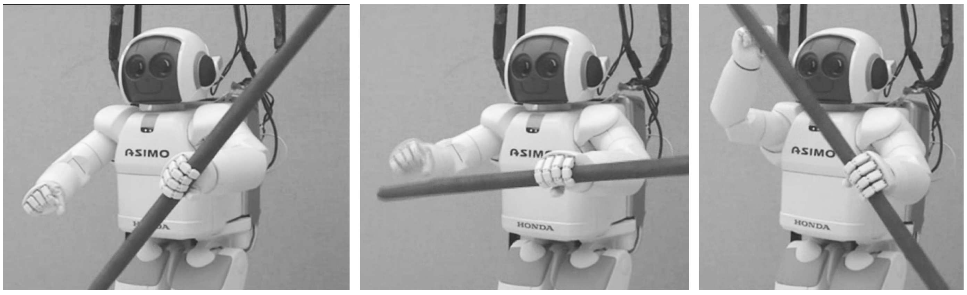


Fig. 14. The robot avoids the rotating bar which is attached to its left hand. Motion targets are given to both the right and left hand, however, the right hand avoids the left hand since the left bar has higher motion priority.

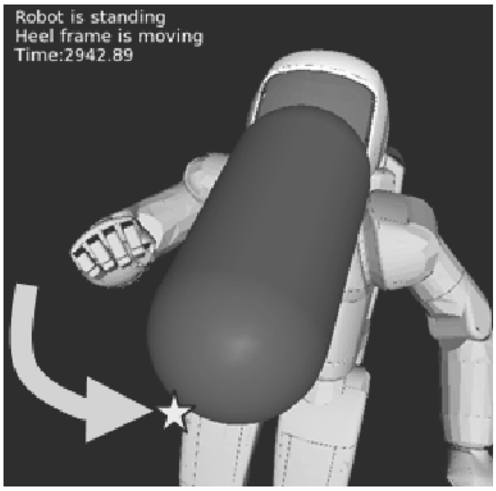


Fig. 15. The virtual object (sphere swept line in front of the robot's head) is generated between the head and the target (star) so that the arms motions don't violate the gaze line.

8.5.2. *A virtual object is avoided to realize occlusion avoidance*

It is also possible to define *virtual objects* and attach them to the body schema. We propose to use this for applications like "occlusion avoidance" as shown in Fig. 15. One of the major problems when the robot grasps an object is occlusions (the hands hide the target object). We defined a virtual segment between the robot's head and the target so that hands do not enter the central field of view. Only just before the robot's hand covers the object, the virtual segment is switched off. By this method, trajectories do not hinder visual tracking of targets.

9. Discussion

The collision avoidance works in different situations, not only when the robot is standing but also during walking. If the target cannot be reached, the robot effectors move to a position which is closest to the target. The collision avoidance motion is composed of two parts: the task space motion of the whole body motion control and the nullspace motion of the collision avoidance control. In particular, the collision avoidance works only in one direction for each arm and the redundancy can be used for target reaching motions in nullspace.

The continuous task priority changing is realized by the distance between closest points pairs in a simple way. Both motions are always working even if one controller does not work. The collision avoidance works not only for robot segments but also for external objects, which can be dynamically modified or switched on and off.

The collision avoidance works with only one DOF in the avoidance direction and other DOFs can be used for other criteria such as a target reacting motion. Therefore, the system can reach given targets faster than the method which uses three DOFs.

9.1. *Coping with planning*

The collision avoidance system is running in the lowest layer of the behavior control system. The modules in the layer react locally to given targets but modules in upper layers should optimize motions more globally. By coping with upper layers' modules, the possible problems which happen in the collision avoidance system can be solved by the system as a whole. For instance, when the robot has to avoid many obstacles or conflicts between the collision avoidance and other limitations, local minima may happen which is shown in Fig. 7(b). All local minima cannot be solved in the reactive collision avoidance system but should be solved on a planning system in upper layers which optimized global trajectories with a slower sampling rate than the low level control one. In other words, planning methods should handle global criteria in stable environments while the real-time collision avoidance assures safety and handles the local criteria in highly dynamic environments by superposing the planned trajectory.

Another example is motion priorities. When the right and left arm get too close and have to avoid each other, the collision avoidance system does not know the priority between right arm and left arm. An upper level module that knows about the task context of the motion can interpret the situation and give a motion priority to the collision avoidance system. Currently, we compute the virtual force vector $\boldsymbol{F}_{virtual}$ only by distances but it can be determined taking into account other criteria, for instance, joint limit avoidance, inertia, viscosity or stiffness of the arm.[23,31]

9.2. *Stability of the system*

Kulic *et al.* analyzed the stability of the collision avoidance for one degree of freedom,[13] unfortunately, for our system it is not feasible to do a formal stability analysis of the whole system with many DOFs.

There are however a number of reasons why the systems shows the good stability we observed in our experiments:

- Our system is a first order system, thus oscillations caused by second order systems such as a spring and damper system can not happen.
- The two controllers are not switched but superposed, thus discontinuities caused by switching do not happen.
- The distances of the closest points are continuous and do not jump except in very complex situations which are close to local minima.

10. Conclusion and Outlook

We realized a reactive collision avoidance system on a humanoid robot running on on-board embedded computers in real-time. It works in dynamic situations in which critical segments are moving such as crossing arms, walking to arbitrary targets and reactive motions. The robot moves to a given target while avoiding collisions. The priority between the target reaching motion and the avoidance motion is changed depending on the distance between closest segments continuously in a simple way. The collision avoidance uses only one DOF to separate critical segments and therefore, other available DOFs can be used by other criteria. The collision avoidance contributes to the hierarchical reactive system. The method can be extended to objects that the robot grasps (body schema) and it can be applied to occlusion problems.

We are going to extend it to handle external obstacles and to deal with an environment with complex obstacles based on the behavior control system. Currently, the system does not consider the dynamics of leg motions and therefore, the leg segments do not avoid collisions. We are going to develop the system that is able to take the leg dynamics into account.

Additionally, we are going to consider irregularities which are caused by unilateral constraints.[19]

Acknowledgments

The authors would like to thank all associates in the Honda Research Institute Europe GmbH for their contributions and support. Also the authors would like to thank all associates in Honda R&D Co., Ltd. who are involved in the development of our humanoid robot ASIMO.

References

1. E. Bicho, P. Mallet and G. Schöner, Using attractor dynamics to control autonomous vehicle motion, *Proceedings of the IEEE Industrial Electronics Society* (1998).
2. O. Brock, O. Khatib and S. Viji, Task consistent obstacle avoidance and motion behavior for mobile manipulation, *Proceedings of the IEEE International Conference of Robotics and Automation* (2002).

3. B. Dariush, M. Gienger, J. Bing, C. Goerick and K. Fujimura, Whole body humanoid control from human motion descriptors, *IEEE International Conference on Robotics and Automation* (2008).
4. M. Deloura, *Game Programming Gems2*. Charles River Media, Inc. (2001).
5. C. Ericson, *Real-Time Collision Detection*, The Morgan Kaufman Publishers (2005).
6. M. Gienger, H. Janssen and C. Goerick, Task-oriented whole body motion for humanoid robots, *Proceedings of the IEEE-RAS International Conference on Humanoid Robots* (2005).
7. C. Goerick, B. Bolder, H. Janssen, M. Geinger, H. Sugiura, M. Dunn, I. Mikhailova, T. Rodemann, H. Wersing and S. Kirstein, Towards incremental hierarchical behavior generation for humanoids, *Proceedings of the IEEE-RAS International Conference on Humanoid Robots* (2007).
8. H. Hanafusa, T. Yoshikawa and Y. Nakamura, Redundancy analysis of articulated robot arms and its utilization for task with priority, *SICE* **19**(5) (1983) 421–426.
9. K. Hirai, Current and future perspective of honda humanoid robot, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (1997).
10. O. Khatib, Real-time obstacle avoidance for manipulations and mobile robots, *The International Journal of Robotics Research* **5**(1) (1986) 90–98.
11. J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba and H. Inoue, Dynamically-stable motion planning for humanoid robots, *Autonomous Robots* **12** (2002) 105–118.
12. J. Kuffner, K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba and H. Inoue, Self-collision detection and prevention for humanoid robots, *Proceedings of the IEEE International Conference on Robotics and Automation* (2002).
13. D. Kulic and E. Croft, Real-time safety for human-robot interaction, *Journal of Robotics and Autonomous Systems* **54**(1) (2006) 1–12.
14. E. Larsen, S. Gottschalk, M. C. Lin and D. Manocha, *Fast Proximity Queries with Swept Sphere Volumes*, Technical Report TR99-018, Department of Computer Science, University of N. Carolina, Chapel Hill. (1999).
15. E. Lengyel, *Mathematics for 3D Game Programming & Computer Graphics Second Edition*, Charles River Media, Inc. (2004).
16. A. Liegeois, Automatic supervisory control of the configuration and behavior of multi-body mechanisms, *IEEE Trans. Systems, Measurement, and Control* **7** (1977) 868–871.
17. T. Lozano-Pérez and M. A. Wesly, An algorithm for planning collision-free paths among polyhedral obstacles, *Communications of the ACM* **22**(10) (1979) 560–570.
18. A. A. Maciejewski and C. A. Klein, Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments, *The International Journal of Robotics Research* **4**(3) (1985) 109–117.
19. N. Mansard and O. Khatib, Continuous control law from unilateral constraints, *IEEE International Conference on Robotics and Automation* (2008).
20. N. Mansard and F. Chaumette, Task sequencing for high level sensor-based control, *IEEE Transactions on Robotics* **23**(1) (2007) 60–72.
21. Y. Nakamura, *Advanced Robotics: Redundancy and Optimization* (Addison Wesley Publishing Company, 1991).
22. Y. Nakamura and H. Hanafusa, Inverse kinematic solutions with singularity robustness for robot manipulator control, *Journal of Dynamic Systems, Measurement, and Control* **108** (1986) 163–171.
23. J.-C. Nebel, Realistic collision avoidance of upper limbs based on neuroscience models, *Computer Graphics Forum* **19**(3) (2000) 219–228.
24. J. Paillard, Body schema and body image — a double dissociation in deafferented patients, *In Motor Control, Today and Tomorrow, eds. G. N. Gantchev, S. Mori*

and J. Massion. Bulgarian Academy of Science Sofia: Academic Publishing House. (1999).

25. L. Sentis and O. Khatib, Synthesis of whole-body behaviors through hierarchical control of behavioral primitives, *International Journal of Humanoid Robotics* (2005).
26. H. Seraji and R. Colbaugh, Improved configuration control for redundant robots, *Journal of Robotic System* **7**(6) (1990) 897–928.
27. F. Seto, K. Kosuge and Y. Hirata, Self-collision avoidance motion control for human robot cooperation system using RoBE, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (2005).
28. B. Siciliano and J.-J. E. Slotine, A general framework for managing multiple tasks in highly redundant robotic systems, *5th International Conference on Advanced Robotics* (1991).
29. H. Sugiura, M. Gienger, H. Janssen and C. Goerick, Real-time self collision avoidance for humanoids by means of nullspace criteria and task intervals, *Proceedings of the IEEE-RAS International Conference on Humanoid Robots* (2006).
30. H. Sugiura, M. Gienger, H. Janssen and C. Goerick, Real-time collision avoidance with whole body motion control for humanoid robots, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (2007).
31. T. Tsuji, P. G. Morasso, K. Goto and K. Ito, Human hand impedance characteristics during maintained posture, *Biological Cybernetics* **72** (1995) 475–485.
32. T. Tsuji, S. Nakayama, A. Araki and K. Ito, Instantaneous inverse kinematics solution for redundant manipulators based on virtual arms and its application to winding control, *JSME International Journal* **38**(1) (1995) 87–93.
33. G. van den Bergen, *Collision Detection in Interactive 3D Environments* (The Morgan Kaufman Publishers, 2004).
34. C. W. Wampler, Manipulator inverse kinematic solutions based on vector formulations and damped least squares methods, *IEEE Trans. Systems, Measurement, and Control* **16**(1) (1986) 93–101.

Hisashi Sugiura studied at University of Tsukuba, Japan and University of Alberta, Canada. He received his BS and MS degrees from University of Tsukuba, in 1991 and 1993, respectively. He then joined Denso Corporation, Japan and developed voice synthesis and voice recognition systems in Institute for Advanced Studies in Artificial Intelligence in Chukyo University, Japan. From 1997, he was Honda R&D Co., Ltd, Japan and developed humanoid robots. Currently he is a senior scientist at the Honda Research Institute Europe, Germany.

Michael Gienger graduated as Diplom-Ingenieur in 1998 at the Technical University of Munich, Germany. From 1998 to 2003, he was research assistant at the Institute of Applied Mechanics of the TUM, addressing issues in design and realization of biped robots. He received his PhD degree in 2003. Currently Michael Gienger is a senior scientist at the Honda Research Institute Europe, Germany. His research interests include mechatronics, robotics, control systems and cognitive systems, with a particular affection for humanoid robots.

Herbert Janßen received his Diploma degree in Physics from the Westphalian Wilhelms University of Muenster, Germany in 1989. From 1989 to 1999 he was researcher at the Institute for Neuroinformatics, University of Bochum, Germany. From 1999 to 2004 he worked in the Honda R&D robot lab in Wako, Japan. Since 2004 he works as a Principal Scientist for the Honda Research Institute Europe, Germany.

Christian Goerick studied Electrical Engineering at the Ruhr-Universität, Bochum, Germany, and at the Purdue University, Indiana, USA. He holds a Doctoral Degree in Electrical Engineering and Information Processing from the Ruhr-University of Bochum. During his time in Bochum he was Research Assistant, Doctoral Worker, Project Leader and Lecturer at the Institute for Neural Computation, Chair for Theoretical Biology. The research was concerned with biologically motivated computer vision for autonomous systems and learning theory of neural networks. Dr. Goerick is currently Chief Scientist at the Honda Research Institute Europe, Germany.